

In the specification

Please amend paragraph [0013] as follows:

In an embodiment of the present invention, object shell console 102 uses attributes of an interpreter programming language, such as the ~~Java~~ JAVA programming language, to connect to an application that is to be analyzed without terminating the application. Once connected, object shell console 102 can provide information to a user to analyze the executing application without terminating or impeding the executing application. Object shell console 102 can provide this capability by connecting to the ~~Java~~ JAVA programming language virtual machine (JVM) on which the application to be analyzed is executing.

Please amend paragraph [0014] as follows:

One exemplary way for making this connection is to use ~~Java~~ JAVA programming language remote method invocation (RMI). ~~Java~~ JAVA programming language RMI is a well-known tool that can be used to access one JVM from another JVM. Using ~~Java~~ JAVA programming language RMI allows remote invocation and execution of applications and methods. To invoke a remote application (or method), ~~Java~~ JAVA programming language RMI creates a thread to the other application or method. Creation of the new thread using ~~Java~~ JAVA programming language RMI occurs without interrupting or terminating the executing application. Thus, the application to which the connection is made continues to execute normally.

Please amend paragraph [0015] as follows:

The application that is connected to and invoked can be either local or remote to administration client 104. Once the connection is established, object shell console 102 has access to the internal structure of the invoked application. This access is allowed by a feature of the Java programming known as introspection. Introspection was developed for ~~Java~~ JAVA programming language Beans to allow for integrated development environments (IDEs) to visually manipulate graphical components to build applications. The classes that are used for introspection are Class, Method and Field. These classes are

also the components of an object. When object shell console 102 retains a reference to a running object by creating the thread, these classes are used to extract the fields and execute the methods of the object. The fields of the object can be manipulated to create different behaviors in the object. Methods can be re-executed with argument values supplied by an operator or system administrator. The values returned by the method's execution can be displayed for analysis.

Please amend paragraph [0022] as follows:

An example will help to clarify the present invention. Assume there is an indication of a performance problem with order database 118. Such indications include errors in error logs corresponding to order database 118 and complaints by users of order database 118. Using object shell console 102, a user of the present invention attaches to order application 110. As described above, this attachment is preferably achieved through creation of a new thread for the application using Java JAVA programming language RMI. The new thread gives object shell console 102 access to the internal structure of order application 10.

Please amend paragraph [0024] as follows:

Figure 2 is a method for analyzing an executing computer application without termination or interruption of the executing application according to the present invention. The method can be executed by an administration client as described above. In step 201, a user selects an application for analysis. In step 202, the present invention attaches to an application to be analyzed. As described above, this connection can be made by creating a thread to the executing application. In one embodiment of the present invention, Java JAVA programming language RMI is used to create the thread.